Vision-based control for space applications

K. Kapellos, François Chaumette, M. Vergauwen, A. Rusconi, L. Joudrier

Vision Based Control for Space Applications

Konstantinos KAPELLOS[(1)], Francois CHAUMETTE[(2)], Maarten VERGAUWEN[(3)], Andrea RUSCONI[(4)], Luc JOUDRIER[(5)]

[(1)]*TRASYS Space, Terhulpsesteenweg 6 C, B-1560 Hoeilaart, Belgium*
Konstantinos.Kapellos@trasys.be

[(2)]*IRISA/INRIA Rennes, Campus de Beaulieu, 35 042 Rennes-cedex, France*
Francois.Chaumette@irisa.fr

[(3)] *K.U.Leuven - ESAT-PSI, Kasteelpark Arenberg 10, B-3001 Heverlee, Belgium*
Maarten.Vergauwen@esat.kuleuven.be

[(4)] *GALILEO AVIONICA, Via Montefeltro, 8, 20156 Milano, Italy*
Andrea.Rusconi@galileoavionica.it

[(5)]*ESA, ESTEC, Keplerlaan 1, 2200 AG Noordwijk, The Netherlands*
Luc.Joudrier@esa.int

Abstract

This paper presents the work performed in the context of the VIMANCO ESA project. It has the objective of improving the autonomy, safety and robustness of robotics system using vision. The approach we propose is based on an up-to-date recognition and 3D tracking method that allows to determine if a known object is visible on only one image, to compute its pose and to track it in real time along the image sequence acquired by the camera, even in the presence of varying lighting conditions, partial occlusions, and aspects changes. The robustness of the proposed method has been achieved by combining an efficient low level image processing step, statistical techniques to take into account potential outliers, and a formulation of the registration step as a closed loop minimization scheme. This approach is valid if only one camera observes the object, but can also be applied to a multi-cameras system. Finally, this approach provides all the necessary data for the manipulation of non cooperative objects using the general formalism of visual servoing, which is a closed loop control scheme on visual data expressed either in the image, or in 3D, or even in both spaces simultaneously. This formalism can be applied whatever the vision sensor configuration (one or several cameras) with respect to the robot arms (eye-in-hand or eye-to-hand systems). The global approach has been integrated and validated in the Eurobot testbed located at ESTEC.

2. Background, Objectives and Overall Approach

Future Space Automation and Robotics applications require the use of vision to perform their calibration and the required precise interactions with the environment. Consequently, vision is compulsory to increase the autonomy of the space robotics agents.

The VIMANCO activity is mainly targeted to EUROBOT, whose purpose is to prepare and assist EVAs on the International Space Station. A typical scenario for the EUROBOT is to place an APFR (Adjustable Portable Foot Restraint) at given locations on the ISS. This involves walking on the handrails and inserting the APFR into a specific fixture called a WIF. In this context, vision is an enabling technology both for the autonomy and the safety of EUROBOT:

- First, although the positions of the handrails and fixtures are well known, there will be some inaccuracy in the placement of the robot, increasing with movements. Vision processing of images would allow the EUROBOT to know the precise positions of the objects to grasp and where to insert or place them. This is a prerequisite to perform the grasping or insertion task itself.
- Second, object recognition would provide the EUROBOT with the ability to check the environment with respect to its a priori knowledge and detect discrepancies. Extending this concept, it would allow the EUROBOT to “know” position of astronauts with respect to itself, representing very

valuable information for advanced safety functionalities.

The European Robotic Arm (ERA) already performs insertion tasks using vision, however, it requires a specific visual target to process the position of the objects to grasp. In the case of the EUROBOT, it is not possible to put a target on every single object. Vision has therefore to cope with non-cooperative objects, i.e. objects that are not equipped with optical markers.

The use of vision in space has to tackle several specific problems and in particular the extreme light difference in images. This means that direct sunlight makes objects appear very bright while shadows are totally dark. Vision algorithms must be very robust in coping with effect of shadows moving in the imaged scene to allow safe and stable manipulation at anytime. Another major space problem is lack of computing power for processing images. Resource (i.e. energy, volume, mass) and environmental constraints (i.e. thermal dissipation, radiation compatibility) limit performance of computers that may be used in space.

In this framework, the main objectives of the VIMANCO activity are first, to define a Vision System Architecture applicable to EUROBOT taking into account the characteristics of the EUROBOT environment and the applicability of the vision techniques to the EUROBOT operations, second to implement a Vision Software Library allowing Vision Control for Space Robots and finally to breadboard the specific HW/SW and to demonstrate it on the ESTEC EUROBOT testbed.

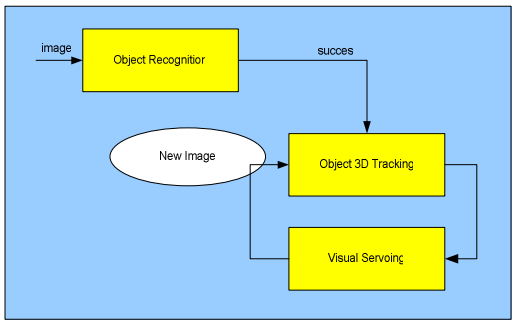


Figure 1: For Vision Based Manipulation of a non-cooperative object three steps are required: Object Recognition, Object Tracking and Visual Servoing

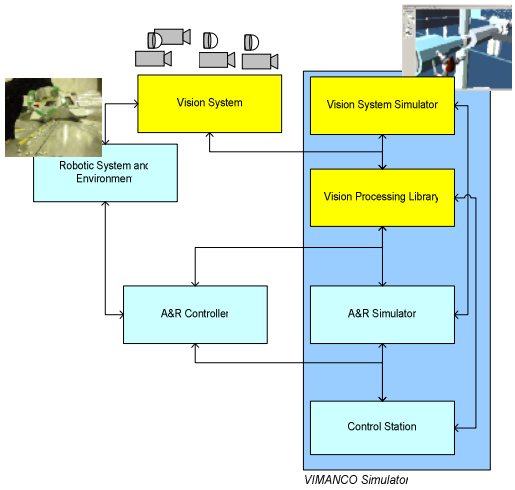


Figure 2: VIMANCO system architecture

To meet the Vision Control objectives, various steps are then required: first the object of interest has to be detected and recognized in the image acquired by the camera. This recognition step must also provide a coarse localization of the object in both 2D and 3D with respect to the camera. Usually this recognition step is time consuming and the result of the localization is not precise enough to be considered for controlling the robot. Therefore we propose to consider a tracking process. Once the object is known, it is possible to track it, over frames and at video rate, using 3D model-based tracking algorithms. These algorithms can use a unique camera but can also consider stereo with small or wide baseline. Finally the output of this algorithm (precise 2D and 3D localization) can be used to control the movement of the robot according to a predefined task. We now describe the three different steps.

Figure 2 illustrates the global VIMANCO system design. It is composed by:

- The *Vision System.* It consists of a stereo pair of cameras attached on a mechanical support and two independent cameras. The stereo camera and each of the two independent cameras dispose an illumination device.
- The *Vision System Simulator*. It is a 3D graphic tool used to reconstruct the robotic system and its environment to produce virtual images. It simulates as faithful as possible the Vision System mounted on the targeted robotic system.
- The *Vision Processing and Object Recognition Library*. It implements all functionality needed for Object Recognition, Object Tracking and Visual Servoing. It provides also the means to control their execution and to communicate with the other systems.

- The *A&R Simulator*. It replaces the robot controller functionality that is needed to validate and demonstrate the whole approach. In real operations the A&R Simulator is replaced by the corresponding *A&R Controller*.
- The *Control Station*. It provides the HMIs that allow the operator to run the VIMANCO simulations. It allows activating Actions/Tasks on the A&R Simulator, to configure, monitor and control the Vision System Simulator and to visualize the acquired images.

3. Object Recognition

The goal of Object recognition is, as clearly stated in the name, the art of finding back specific objects when seen in new situations or different images. It is a subject that has been studied in computer vision since its early days (about fifty years ago). Most systems in those days worked rather ad-hoc on simplified objects such as polygons and polyhedrons. The research community has come a long way since then. In general the task of recognizing an object is made difficult because of the possible variability of the camera's internal parameters, its position and orientation, the illumination conditions and even the constellation of the visible objects.

In the VIMANCO activity Object Recognition has been designed and implemented as follows:

- During the mission preparation phase, the off-line training performed in which the objects to be recognized are modeled using features and their corresponding feature descriptors and 3D coordinates
- During operations, the on-line object recognition is based on the same procedure of feature extraction and description and an additional feature matching and verification step.

During the off-line training the Object Recognition component describes the objects to be recognized using local invariant features. A specific application with a HMI front-end is employed for this. The algorithm and corresponding data flow of this application are shown in Figure 3.

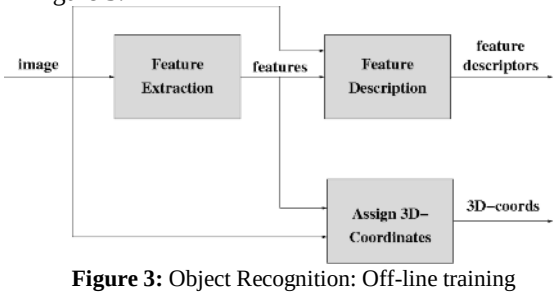


Figure 3: Object Recognition: Off-line training

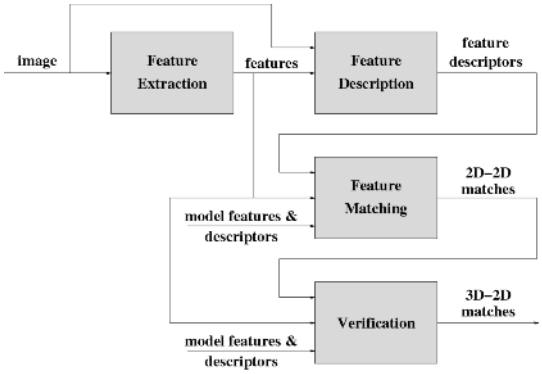


Figure 4: On-line Object Recognition

The input to the system consists of images of the object to be modelled. For every image a set of features is extracted first, using the Feature Extraction component. These will typically be affine invariant or rotation-scale invariant features like MSER, IBR, SIFT or SURF regions. When these features are extracted, a feature descriptor can be computed for each of them. The Feature Description component is used to this end. The output consists of a feature descriptor for each feature, containing the description of this feature.

In order to initialize the camera pose for Object Tracking, 3D-2D correspondences will be computed later. These can be used to compute the camera pose w.r.t. the object. Since feature matching is performed in the images, we need to assign 3D coordinates to every feature. We do so using a specifically dedicated graphical tool to Assign 3D-Coordinates.

As depicted in Figure 3, the data-flow between the 3 components of the off-line training step is straightforward. An image of the object is the only input of the system. The location of the features is an extra input for the description phase. In order to compute the 3D coordinates of the features, a (simplified) 3D model of the object is needed as well.

During real operation, the system needs to identify objects in the image or certify their presence. This is the goal of the object recognition phase, implemented in the Object Recognition activity. The data flow diagram of this activity is shown in Figure 4. We recognize the first two components of this phase. The Feature Extraction and Feature Description components are identical to those in the off-line training phase. Indeed, the first step in recognizing an object in an image consists of locating features in this image and describing these features using the same algorithm as before. The newly found feature descriptors can then be matched to the feature descriptors of the objects in the database. This is done in the Feature Matching component. The result of this

component consists of matches between features, i.e. 2D-2D correspondences. These results can contain mismatches, while other (correct) matches might have been missed. This can be ameliorated by the Verification component, which will output its result in the form of matches between 3D coordinates (found by the Assign 3D Coordinates component in the pre-processing phase) and 2D coordinates (of the features extracted in the target image. These 3D-2D correspondences can be used to compute an initialization of the camera pose w.r.t. the object.
The data flow is clear from Figure 4. Features and feature descriptors are extracted from the target image and are matched to the model features, i.e. the features extracted in the model images during the off-line training phase. The resulting 2D-2D correspondences are checked in the verification step to yield 3D-2D correspondences.

4. Object Tracking

Elaboration of object tracking algorithms in image sequences is an important issue for applications related to robot vision based control or visual servoing and more generally for robot vision. A robust extraction and real-time spatio-temporal tracking process of visual cue is indeed one of the keys to success of a visual servoing task. To consider visual servoing this spatial robotics context, it is fundamental to handle "natural" scenes without any fiducial markers but with complex and non cooperative objects in various illumination conditions. The goal of object tracking is then to determine the position in every image acquired by a camera of particular object (which has been previously recognized). This position may be defined in the image space (we then have a 2D tracking algorithm) or in 3D with respect to the camera or to a world frame (we then have a 3D tracking algorithm). Note that when a 3D localization is available, then the 2D position is also available.

In our work 3D model-based tracking is used since it is usually more robust and it is then more suitable for the considered application. Furthermore such algorithm provides both 2D and 3D localization of the tracked object and it is then very suitable for any kind of vision-based control algorithms.

In particular, the Object Tracking component allows the localisation, at video rate, of a given object, by using for each frame one (or more) current image(s) of this object acquired by one (or more) camera(s), a CAD model of the considered object and its previous localisation. The pose estimation is based on the robust virtual visual servoing technique in which the visual features are the distances between the object contour and the current set of extracted points. In practice, a virtual camera is moved from the previously determined pose to a pose where the projected contour of the object matches the set of extracted points. At convergence, the current pose of the camera gives the pose of the object.

The used control law is very similar to the one used in the Visual Servoing component, excepted that a robust estimator is directly included into the control law in order to correctly reject potential outliers and to estimate the pose of the tracked object with a good precision.

Figure 5 illustrates the block diagram of the Object Tracking algorithm.

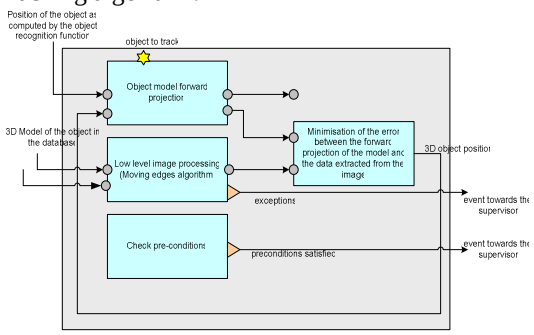


Figure 5: Object Tracking component

5. Visual Servoing

Basically, vision-based robot control or visual servoing techniques consist in using the data provided by one or several cameras in order to control the motions of a dynamic system. Such systems are usually robot arms, or mobile robots, but can also be virtual robots, or even a virtual camera. A large variety of positioning tasks, or mobile target tracking, can be implemented by controlling from one to all the n degrees of freedom of the system. Whatever the sensor configuration, which can vary from one on-board camera on the robot end-effector to several free-standing cameras, a set of k measurements has to be selected at best, allowing controlling the m degrees of freedom desired. A control law has also to be designed so that these measurements s(t) reach a desired value s^*, defining a correct realization of the task. A desired trajectory $s^*(t)$ can also be tracked. The control principle is thus to regulate to zero the error vector $s(t)-s^*(t)$. With a vision sensor providing 2D measurements, potential visual features are numerous, since as well 2D data (coordinates of feature points in the image, moments, ...) as 3D data provided by a localization algorithm exploiting the extracted 2D features can be considered. It is also possible to

combine 2D and 3D visual features to take the advantages of each approach while avoiding their respective drawbacks. Figure 6 illustrates the block diagram of the Visual Servoing algorithm.

If the task is specified as a 3D displacement in the robot end-effector frame (called after the hand), or as a pose between the hand or the camera and the observed object, an accurate calibration of the camera and of the eye-hand pose has to be performed, so that the task can be expressed as an accurate pose to reach between the camera and the object.

A coarse camera and eye-hand calibration is sufficient in the case where the task is specified as a particular position of the object in the image. In practice, this can be obtained using an off-line teaching by showing step where the end-effector is moved once at its desired position with respect to the object and the corresponding image is stored. In that case, the data extracted from the vision sensor will be biased due to the calibration errors, but the robustness of the visual servoing with respect to calibration errors will allow to move accurately the arm so that the final image corresponds to the desired one, ensuring a correct realization of the task.

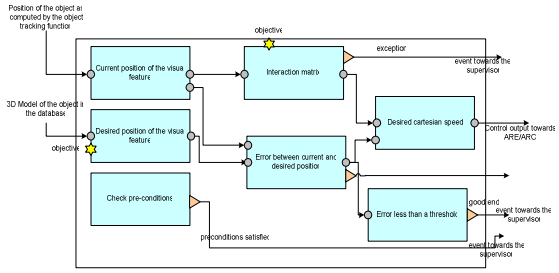


Figure 6: Visual Servoing component

6. The Vision System

The Vision System supports the characterisation of the object-recognition and the visual servoing algorithms developed in this activity. Since the system is meant as a tool for EUROBOT, the Vision System mimics the EUROBOT setup.

We consider a camera-setup as shown in Figure 7: a stereo pair of digital cameras attached on a mechanical support and two independent digital cameras to be attached to the end effector of two of the EUROBOT testbed arms. In order to match as perfectly as possible the ideal illumination characteristics of the cameras, the stereo pair and each independent camera will be provided with an individually regulated illumination sub-system, consisting of hallogen head-lights.

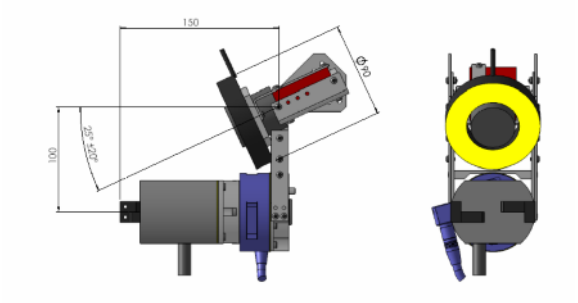


Figure 7: The VIMANCO arm camera vision system

7. Vision System Simulator

The development and the validation of the vision algorithms that implement the previous objectives requires images at each new robot position as input and a robot to execute the required control output. Disposing such a hardware configuration for the development and the first tuning of the algorithms is impracticable since very time consuming: a vision simulator that provides the possibility first to produce realistic virtual images from a synthetic environment and second to control a simulated robot in this environment has been developed and integrated for testing and tuning the vision algorithms.

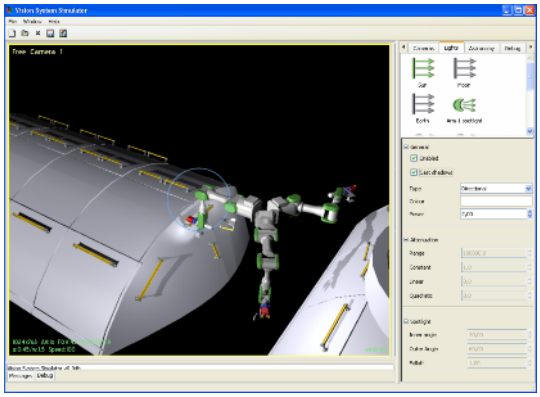


Figure 8: The VIMANCO Vision System Simulator HMI

In particular the VIMANCO Vision System Simulator provides the means (see Figure 8 and Figure 9):

- To model in 3D the elements of a robotised cell. It includes the robots and payloads to be manipulated and the models of vision sensors (cameras and stereo head). Images quality parameters associated to a camera are adjustable, e.g. noise, distortion, glare for testing in various conditions. Lighting sources associated to cameras but also celestial objects (the sun and the moon) are modelled as well.

- To specify the ambient conditions allowing to consider and to adjust direct sunlight intensity and direction, ambient light and surface reflexivity characteristics.
- To control the movement of a robot, and so of the attached cameras, based on external inputs.
- To provide realistic images to external systems in real-time compatible with the temporal constraints imposed by the application of vision based control to move a robot.

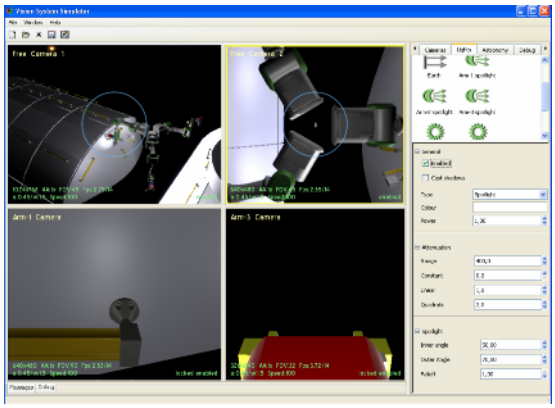


Figure 9: User selected cameras views may be displayed in parallel with the robotic cell free view

The main visualisation technologies and tools used for the Vision Simulator include material and compositor scripting, advanced scene manager, meshes support, resources manager and XML loaders to ease the configuration.

The Material Scripting component is used to declare and maintain material assets outside the Vision System Simulator source code. Each material has a unique name and can be assigned to any surface in the scene when it is modelled using an offline tool (i.e. 3D Studio Max). In the rendering loop the Vision System Simulator parses the material script attached to each polygon and executes the instructions given in it to determine the final surface properties. Material scripting supports:

- Vertex and fragment programs (shaders), both low-level programs written in assembler, and high-level programs written in Cg, DirectX9 HLSL, or GLSL and provides automatic support for many commonly bound constant parameters like worldview matrices, light state information, object space eye position etc.
- The complete range of fixed function operations such as multi-texture and multi-pass blending, texture coordinate generation and modification, independent color and alpha operations for non-programmable hardware or for lower cost materials.
- Multiple pass effects, with pass iteration if required for the closest 'n' lights.
- Multiple material techniques with automatic fallback in case the best technique is unsupported by the used hardware.
- Material LOD.
- Load textures from PNG, JPEG, TGA, BMP or DDS files, including unusual formats like 1D textures, volumetric textures, cubemaps and compressed textures (DXT/S3TC)

The Scene Manager component is in charge of the contents of the scene which is to be rendered. It is responsible for organising the contents using whatever technique it deems best, for creating and managing all the cameras, movable objects, lights and materials (surface properties of objects), and for managing the 'world geometry' which is the sprawling static geometry usually used to represent the immovable parts of a scene. Scene manager features include:

- Highly customisable, flexible scene management not tied to any single scene type.
- Hierarchical scene graph; nodes allow objects to be attached to each other and follow each others movements, articulated structures etc.
- Multiple shadow rendering techniques, both modulative and additive techniques, stencil and texture based, each highly configurable and taking full advantage of any hardware acceleration available.
- Scene querying features

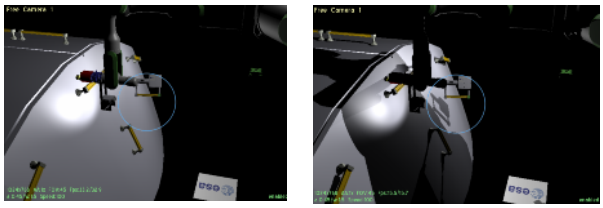


Figure 10: a) Cast shadows off all lights, spotlight power 1.4 b) Cast shadows on the spot light

9. Experimental results

The performance of the Object Recognition, the Object Tracking and the Visual Servoing have been extensively tested using a classical 6-axis robot at IRISA-INRIA Rennes ([6]) and the Eurobot testbed located at ESTEC (see Figure 11) with respect to different camera positions, illumination conditions and occlusions. We present here after the results on two objects: a mockup of the Articulated Portable Foot Restraint (APFR) and a mockup of a handrail.

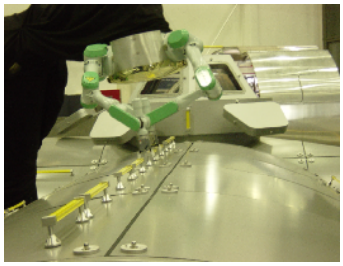


Figure 11: The Eurobot testbed at ESTEC

Figure 12 presents model images of the APFR used to train the Object Recognition system. These images have been taken from one position but with different illumination conditions.

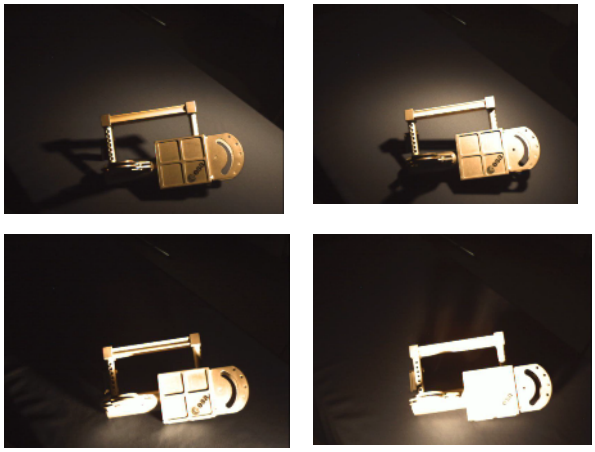


Figure 12: Object Recognition: model images taken from the same position with different illumination conditions

Figure 13 shows the modeling process with one of these images. SURF features are found and described in the image. The 3D coordinates of these features are computed by registering a 3D model of the object with the current viewpoint and reading out the Z-buffer. When features are computed for all model images, these features are combined into a single image, shown in Figure 14.

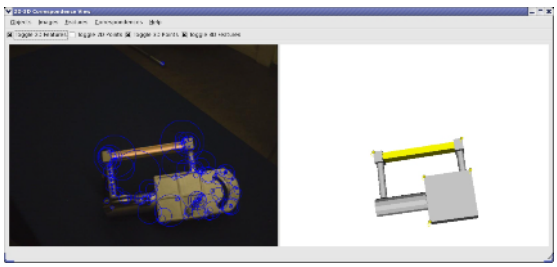


Figure 13: The process of modeling an image, using SURF features and registration of a 3D model to compute the 3D coordinates of the features

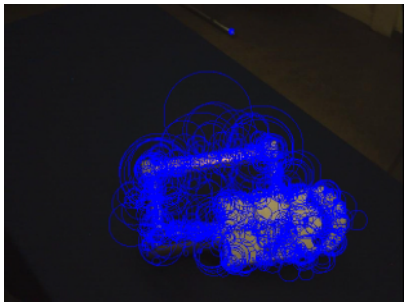


Figure 14: One model image holds the combination of all features, computed from all illumination conditions.

During experiments, at different initial robot positions images where taken from a substantially different viewpoint and different illumination conditions (see Figure 15).

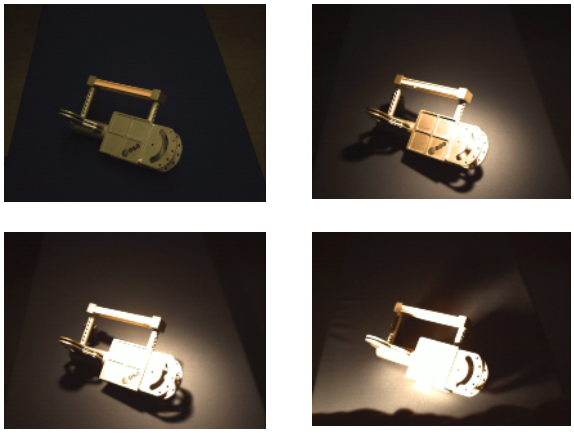


Figure 15: Images taken from different viewpoints with different illumination conditions

It should be noted that the employed SURF features are, while fast to extract and describe, only invariant under rotation and scale changes. The substantial out-of-plane rotation that can be seen in the images is mathematically not covered by the features but, the experiments showed that a reasonable amount of robustness of the features still allows the images to be matched.

For the first three images of Figure 15 the object position has been computed with a sufficient precision to allow the initialization of the Object Tracking/Visual Servoing. We used the combined model image to match against. In the employed strategy, matches are searched between model and target image and then only the matches are kept that are consistent with the 3D model of the object. This consistency is computed with a camera-RANSAC algorithm that computes the pose of the camera as well. Figure 16 shows the inliers

to the RANSAC algorithm for the second image. The resulting pose is illustrated in Figure 17.

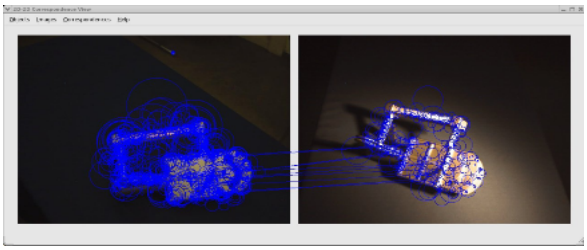


Figure 16: Inliers of the camera-RANSAC for the second image using relative matching

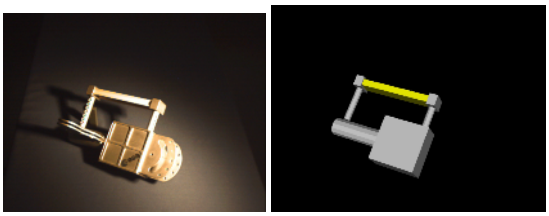


Figure 17: Original image and virtually rendered image computed by the camera-RANSAC algorithm

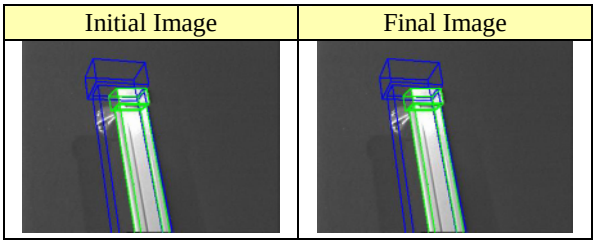
The computation of the position of the APFR in the fourth image of Figure 15 was not possible because the object is overexposed due to the reflection of the metal.

Object Tracking and Visual Servoing are tested performing five positioning tasks toward the same desired position under different initial positions and illumination conditions. For each positioning task we repeated the positioning five times in order to check experiment repeatability. Measurements are provided using the INRIA Afma Robot which precision is 0.1cm for translation and 0.5 degree in rotation.
This robot is able to provide the position of the camera in the robot reference frame that is given by the homogeneous matrix ${}^{c}M_{f}$.
Accuracy measurements are then handled in two step:

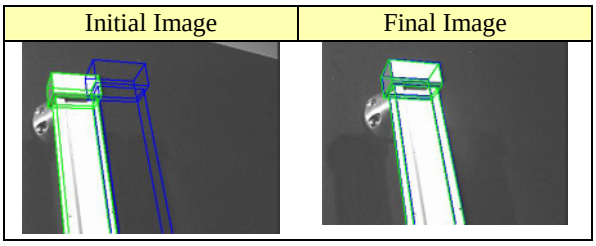
- Measurement of the desired camera 3D position ${}^{cd}M_{f}$.
- Measurement of the camera final position ${}^{c}M_{f}$
- The accuracy is then given by ${}^{cd}M_{c} = {}^{cd}M_{f}\ {}^{f}M_{c} = {}^{cd}M_{f}\ {}^{c}M_{f}^{-1}.{}^{cd}M_{c}$ actually measures the position of the final camera position with respect to the desired one.

The results of two experiments are illustrated hereafter:

	Tx	Ty	Tz	Rx	Ry	Rz
init	0.389	0.147	0.4427	0.4032	0.4493	0.116
Exp1	-0.0018	0.001	-0.0004	0.003	0.007	-0.002
Exp2	-0.0021	0.000	0.0002	0.002	0.007	-0.003
Exp3	-0.0018	0.001	-7.2e-06	0.004	0.007	-0.002
Exp4	-0.0016	0.0008	-5.6e-07	0.002	0.006	-0.002
Exp5	-0.0022	0.0010	-1.8e-05	0.003	0.008	-0.003



	Tx	Ty	Tz	Rx	Ry	Rz
init	0.4647	0.12104	0.5001	0.403	0.449	0.116
Exp1	-0.0122	-0.0248	0.0032	-0.088	0.045	-0.012
Exp2	-0.0077	-0.0079	0.0037	-0.023	0.025	-0.009
Exp3	-0.0099	-0.0073	0.0038	-0.020	0.033	-0.011
Exp4	-0.0090	-0.0059	0.0034	-0.017	0.031	-0.010
Exp5	-0.0104	-0.0086	0.0039	-0.024	0.034	-0.012



For additional experimental results with the APFR we refer the reader to [6] while videos are available to the INRIA - Lagadic site (http://www.irisa.fr/lagadic).

10. References

[1] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, 8(3):313–326, June 1992.

[2] E. Malis, F. Chaumette, and S. Boudet. 2 1/2 D visual servoing. *IEEE Trans. on Robotics and Automation*, 15(2):238–250, April 1999.

[3] A.I. Comport, E. Marchand, and F. Chaumette. Robust model-based tracking for robot vision. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'04*, volume 1, pages 692–697, Sendai, Japan, September 2004.

[4] T. Tuytelaars, L. Van-Gool, L. Dhaene, and R. Koch, "Matching affinely invariant regions for visual servoing", Proceedings of the IEEE Conference on Robotics and Automation, pages 1601-1606, 1999.

[5] V. Ferrari, T. Tuytelaars, and L. Van-Gool, "Simultaneous object recognition and segmentation by image exploration" In Proceedings of the European Conference on Computer Vision, 2004.

[6] F. Dionnet, E. Marchand: "Robust Model-Based Tracking with Multiple Cameras for Spatial Applications", ASTRA 2006.